Dynamic Controllability with Single and Multiple Indirect Observations

Paul Morris
NASA Ames Research Center
Moffett Field, CA 94035, U.S.A.

Arthur Bit-Monnot
University of Sassari
Sassari, Italy

Abstract

A recent paper introduced a transformation-based approach for determining dynamic controllability of Simple Temporal Networks with Uncertainty (STNUs) extended to have variably-delayed observations of uncontrolled timepoints. Although the approach correctly determines dynamic controllability, it does not always provide the most flexible possible dynamic strategy. We show how to refine the approach in a way that improves the flexibility, and further extend it to a class of Partially Observable STNUs where the hidden timepoints can be indirectly observed via a chain of contingent links. We show how to construct a labeled distance graph for these problems, leading to a complete solution. This approach handles "single-headed" chained contingent links. For "multi-headed" problems, we prove a theorem characterizing their dynamic controllability in isolation. This provides a check on more general networks (and more general methods). We also consider potential extensions of the single-headed approach to multi-headed problems and point out some difficulties that arise.

Introduction

The Simple Temporal Network (STN) formalism introduced by Dechter, Meiri, and Pearl (Dechter, Meiri, and Pearl 1991) has proved very fruitful for reasoning about temporal plans. It has been extended in various directions including the STNU formalism that deals with external events whose timing is uncertain (Vidal and Fargier 1999; Morris, Muscettola, and Vidal 2001; Hunsberger 2009; Morris 2014), and effective algorithms have been developed to handle these problems. An additional extension (Moffitt 2007) introduced the Partially Observable STNU (POSTNU) formalism that may include uncontrolled timepoints that can be observed only indirectly, through their subsequent effects on other timepoints that are observable. In this paper, the uncontrolled timepoints that cannot be directly observed may conveniently be called *hidden* timepoints.

A recent paper (Bhargava, Muise, and Williams 2018) introduces a related extension of STNU, called Variable Delay STNU, where an uncontrollable event is determined to have occurred only after some delay whose duration is itself uncertain. It was noted that a Variable Delay STNU can be modeled as a special case of a POSTNU where the observational delay is represented as a separate contingent link that is activated by the uncontrollable event that is being observed. Thus, viewed as a POSTNU, the network involves two chained contingent links. The uncontrolled timepoint that is observed only indirectly (via the chained link) is a hidden timepoint. The Variable Delay paper considers networks that may have additional requirement constraints on the hidden timepoints. It introduces novel methods to tranform such Variable Delay STNUs into an STNU where contingent timepoints are either instantaneously observable or never observable. The result is essentially a POSTNU without chained constraints, which allows dynamic controllability to be determined by existing methods (Bit-Monnot, Ghallab, and Ingrand 2016). It also presents an execution strategy for networks that are found to be dynamically controllable.

The method presented in the Variable Delay paper correctly determines dynamic controllability and presents a valid execution strategy. However, the strategy presented is not always the most flexible possible, as we will show by an example. This can be remedied by a "doubling" strategy where the timepoint following a hidden timepoint is split in two. We will show how the doubling arises naturally in a first principles analysis. Also, for a Variable Delay STNU viewed as a POSTNU, a hidden timepoint activates at most one chained constraint. This may be described as a "single-headed" chained constraint. We will consider "multi-headed" cases where several contingent links are activated by the same hidden timepoint, and prove a theorem relating the dynamic controllability of a multi-headed network fragment to a relationship between the "slack" (upper bound minus lower-bound) of the requirement and contingent constraints involved. This result can be used as a check on the validity of transformation methods. Our results extend the set of cases where algorithms for DC checking of POSTNU are complete.

STNUs and Extensions

A Simple Temporal Network (STN) (Dechter, Meiri, and Pearl 1991) is a graph in which the edges are annotated with upper and lower numerical bounds. The nodes in the graph represent temporal events or *timepoints*, while the edges (refered to as *links*) correspond to constraints on the durations between the events. For instance a link $A \xrightarrow{[2,5]} B$ imposes that at least 2 time units and no more than 5 time units elapse between the occurrence of A and the occurence of B. Each STN is associated with a *distance graph* where each link A $\xrightarrow{[x,y]}$ B is replaced by two edges A $\xrightarrow{y}$ B and A $\xleftarrow{-x}$ B. An STN is consistent if and only if the distance graph does not contain a negative cycle.

A Simple Temporal Network With Uncertainty (STNU) is similar to an STN except the links are divided into two classes, *requirement links* and *contingent links*. Requirement links are temporal constraints that the agent must satisfy, like the links in an ordinary STN. Contingent links may be thought of as representing causal processes of uncertain duration, or periods from a reference time to exogenous events; their finish timepoints, called here *contingent timepoints*, are controlled by Nature, subject to the limits imposed by the bounds on the contingent links. All other timepoints, called *executable timepoints*, are controlled by the agent, whose goal is to satisfy the bounds on the requirement links. The start timepoint of a contingent link is called its *activation timepoint* and can be either contingent or executable. Each contingent link is required to have non-negative (finite) upper and lower bounds. An STNU may be thought of as determining a family of STNs where the contingent links take on each of their possible durations; the individual STNs in the family are called *projections*.

In STNUs, the uncontrolled timepoints are assumed to be either all unobservable or all observable, giving rise to different execution strategies. An STNU is *Strongly Controllable* if there is a single schedule that satisfies the requirements in *all* of the projections. An STNU is said to be *Dynamically Controllable* if there is a strategy for scheduling each executable timepoint that depends only on observations that are available (in the past) at the time it is scheduled. Whether an STNU is Dynamically Controllable or not can be determined by an algorithm that runs in cubic time (Morris 2014). The algorithm tightens some constraints in a way that makes explicit limitations on the execution strategies due to the presence of contingent links.

Some of the tightenings involve a temporal constraint called a *wait*. Given a contingent link AB and another link AC, a wait indicates that execution of the timepoint C is not allowed to proceed until after either B has occurred or some specified amount of time t has elapsed since A occurred. More precisely, it corresponds to the constraint $C - A \geq \min(B - A, t)$. Note that a wait reduces to an ordinary temporal constraint in a projection, since there the value of $B - A$ is fixed.

As mentioned, an STN has an alternative representation as a *distance graph* (Dechter, Meiri, and Pearl 1991). Similarly, there is a representation for an STNU called the *labeled distance graph* (Morris and Muscettola 2005) In the labeled distance graph, each requirement link A $\xrightarrow{[x,y]}$ B is replaced by two edges A $\xrightarrow{y}$ B and A $\xleftarrow{-x}$ B, just as in an STN. For a contingent link A $\overset{[x,y]}{\Longrightarrow}$ B, we have the same two edges A $\xrightarrow{y}$ B and A $\xleftarrow{-x}$ B, but we also have two additional edges of the form A $\xrightarrow{b:x}$ B and A $\xleftarrow{B:-y}$ B. These are called *labeled edges* because of the additional "b:" and "B:" annotations indicating the contingent timepoint B with which they are associated. Note especially the reversal in the roles of x and y in the labeled edges. We refer to A $\xleftarrow{B:-y}$ B and A $\xrightarrow{b:x}$ B as *upper-case* and *lower-case* edges, respectively. Observe that the upper-case labeled weight B:-y gives the value the edge would have in a projection where the contingent link takes on its maximum value, whereas the lower-case labeled weight b:x corresponds to the contingent link minimum value. An upper case edge A $\xleftarrow{B:-t}$ C is also used to represent the wait involving A,B,C considered earlier; it is consistent with the *lower* bound on AC that would occur in a projection where the contingent link has its maximum value.

A POSTNU (Moffitt 2007) is essentially an STNU that has both observable and unobservable (hidden) timepoints. Thus, the controllability problem for a POSTNU may be regarded as a combination of Strong and Dynamic Controllability. Moffitt's algorithm for checking the controllability of a POSTNU is complete but not sound in that it might incorrectly label a POSTNU as controllable. Another algorithm, also relying on the compilation to STNUs is provided by (Bit-Monnot, Ghallab, and Ingrand 2016) that is sound but only complete for a subclass of POSTNUs. A polynomial sound and complete algorithm for assessing the controllability of general POSTNU remains to be found. It is **important to note** one particular point with respect to the semantics. A contingent link may be activated by a hidden timepoint. In that case, if the endpoint is observable, the POSTNU semantics specifies that when it is observed, we learn only the *time* of the endpoint, not the *duration* of the link that was activated by the hidden timepoint. Of course we do learn (or can easily calculate) the time difference between the observed endpoint and any previous known time. Other semantics are possible, and may be useful in some applications, but will not be considered in this paper.

Variable Delay

The Variable Delay STNU (Bhargava, Muise, and Williams 2018) formalism is an STNU extension that relaxes the condition of instantaneous observation of contingent timepoints. In this case, the end of a contingent link is not directly observed; instead after some bounded delay (with upper and lower bounds), it is learned that the contingent timepoint has occurred. The duration of the delay is not observed, so the time at which the contingent timepoint occurred is not directly known. However, bounds on the time of occurrence can be inferred from the other observations.

A Variable Delay STNU may be regarded as a special case of a POSTNU where the original contingent link is chained with a separate contingent link that represents the delayed observation process. The original contingent timepoint is treated as hidden. An example is shown in figure 1. Here XE represents the original contingent link, E is hidden, and EY represents the delayed observation. The link EZ is a requirement that is imposed on the hidden timepoint. In the example, X and Z are executable timepoints and Y is an observable timepoint.

Note that the semantics of Variable Delay STNU implies that Y is a "terminal" timepoint, i.e., the corresponding

$$\begin{tikzcd}
\mathrm{X} \arrow[r, Rightarrow, "{[0,5]}"] & \mathrm{E} \arrow[r, Rightarrow, "{[5,10]}"] \arrow[d, "{[0,10]}"] & \mathrm{Y} \\
 & \mathrm{Z} &
\end{tikzcd}$$

Figure 1: Variable Delay as POSTNU

$$\begin{tikzcd}
\mathrm{X} \arrow[r, Rightarrow, "{[a,b]}"] & \mathrm{E} \arrow[r, Rightarrow, "{[g^-,g^+]}"] \arrow[d, "{[u,v]}"] & \mathrm{Y} \\
 & \mathrm{Z} &
\end{tikzcd}$$

Figure 2: Generic Variable Delay

POSTNU may not impose any requirements on the timepoint representing the delayed observation, and it may not activate a new contingent link. In addition, the delayed observation is "single-headed" in the sense that the POSTNU can have only one contingent link that is activated by the hidden timepoint. In this paper, we will develop solution methods that encompass a wider (though still limited) range of problems.

We now review a somewhat simplified description of the Variable Delay solution procedure (Bhargava, Muise, and Williams 2018), as expressed in terms of a POSTNU, for a generic example (figure 2) that parallels the one used in the Variable Delay paper. For the following discussion, as a notational convenience, we define slack(AB) = q-p for any link AB with bounds [p,q].

The solution procedure starts by checking whether slack(XE) $\leq$ slack(EY). If so, it replaces EY by an infinite delay, which essentially discards the EY observation from the POSTNU.

Otherwise it applies the transformations in table 1, which effectively moves the requirement from the unobservable E to an observable Y' as indicated in figure 3. We have rewritten Y as Y' because, as we will see, it is not really the same timepoint as Y. (More on this later.)

The iterated transformation process converts a Variable Delay problem into one in which timepoints are either unobservable or have zero delays. This is essentially a POSTNU in which all the activation timepoints are observable. These are problems for which dynamic controllability can be checked by previous methods (Bit-Monnot, Ghallab, and Ingrand 2016).

Original edges	Replacement edge
$X \overset{[a,b]}{\Longrightarrow} E \overset{[p,q]}{\Longrightarrow} Y$	$X \overset{[a+q,b+p]}{\Longrightarrow} Y'$
$Z \xleftarrow{[a,b]} E \overset{[p,q]}{\Longrightarrow} Y$	$Z \xleftarrow{[a-p,b-q]} Y'$

Table 1: Variable Delay transformations involving a hidden timepoint E

$$\begin{tikzcd}
\mathrm{X} \arrow[r, Rightarrow, "{[a+g^+,b+g^-]}"] & \mathrm{Y'} \arrow[d, "{[u-g^-,v-g^+]}"] \\
 & \mathrm{Z}
\end{tikzcd}$$

Figure 3: Transformed Variable Delay

$$\begin{tikzcd}
\mathrm{X} \arrow[r, Rightarrow, "{[0,5]}"] & \mathrm{E} \arrow[r, Rightarrow, "{[0,1000]}"] \arrow[d, "{[0,10]}"] & \mathrm{Y} \\
 & \mathrm{Z} &
\end{tikzcd}$$

Figure 4: Suboptimal Strategy Example

The Variable Delay paper presents arguments that the transformed problem is dynamically controllable if and only if the original is also, which extends dynamic controllability checking to a wider class of problems.

The paper also presents an execution strategy for the transformed problem. The timepoint designated Y' in figure 3 is treated as though it corresponds to an observation of

$$(\mathrm{t} \geq \mathrm{a} + \mathrm{g}^+) \wedge (\mathrm{Y} \vee \mathrm{t} \geq \mathrm{b} + \mathrm{g}^-)$$

where t is the time as measured since X was executed. That is, if Y was observed earlier than time $\mathrm{a} + \mathrm{g}^+$, then Y' is considered to be observed at time $\mathrm{a} + \mathrm{g}^+$. If Y is observed between time $\mathrm{a} + \mathrm{g}^+$ and time $\mathrm{b} + \mathrm{g}^-$, then Y' is observed when Y is observed. If Y is not observed until after time $\mathrm{b} + \mathrm{g}^-$, then Y' is considered to be observed at time $\mathrm{b} + \mathrm{g}^-$. We say Y' is an observable *derived* from Y.

Improved Dynamic Strategy

The Variable Delay paper does correctly determine the dynamic controllability of a Variable Delay problem (as we will later confirm by a different analysis), and it does present a valid dynamic strategy. However, the dynamic strategy obtained is not always the most general possible (in the sense of preserving the greatest flexibility). Consider the example in figure 4. The Variable Delay procedure would essentially discard the Y observation as one that is "highly uncertain," and treat E as totally unobservable. Then, from XE and EZ, we infer an XZ requirement of [5,10]. Notice however that with the original network, if Y is observed at any time in [0,5] we can immediately infer that E has happened, and so it is safe to go ahead with Z. If Y is not observed, we can nevertheless proceed with Z in [5,10]. This is more flexible than [5,10] only.

As another example, suppose the EZ link had bounds [990,1000] instead. Compiling away E would then impose an XZ requirement of [995,1000]. However, if Y has not finished at time 1000, it is nevertheless safe in the original network to hold off on executing Z until Y finishes (since E must still be within the allowed range), which is more flexible and potentially might not happen until 1005 after X.

$$\begin{tikzcd}
& \mathrm{X} \arrow[ld, "\mathrm{y}:(\mathrm{a}+\mathrm{g}^{+})"'] & \\
\mathrm{Y'} \arrow[rd, "\mathrm{v}-\mathrm{g}^{+}"'] & & \mathrm{Y''} \arrow[lu, "Y:-(\mathrm{b}+\mathrm{g}^{-})"'] \\
& \mathrm{Z} \arrow[ru, "-(\mathrm{u}-\mathrm{g}^{-})"'] &
\end{tikzcd}$$

Figure 5: Doubled Y Timepoint

For execution purposes, discarding the Y observation is overly drastic since it can make a contribution to the dynamic strategy even though it is "highly uncertain." However, if the Variable Delay paper did not perform this preliminary step when slack(XE) $<$ slack(EY), then the first transformation in table 1 could produce a paradoxical contingent link where the lower bound is greater than the upper bound. (Note $(\mathrm{b}+\mathrm{g}^{-})-(\mathrm{a}+\mathrm{g}^{+})=(\mathrm{b}-\mathrm{a})-(\mathrm{g}^{+}-\mathrm{g}^{-})$.)

As it turns out, there is an alternative way of resolving this issue that does not require discarding the Y observation. The basic idea is to replace Y by *two* new timepoints Y' and Y", in which we separate the upper and lower bounds. Otherwise, the transformation is essentially the same as in the Variable-Delay paper. Here, we just indicate how this resolves the flexibility issue; later on, we will show how these timepoints arise in a principled analysis.

Figure 5 shows the transformed network as a labeled distance graph. This is semi-reducible if either Y'$\to$Z or Y'$\to$Z$\to$Y" is negative, i.e., if either $\mathrm{v} < \mathrm{g}^{+}$ or $(\mathrm{v}-\mathrm{u}) < (\mathrm{g}^{+}-\mathrm{g}^{-})$. We then have a semi-reducible negative cycle if the whole cycle is negative, i.e., if $(\mathrm{v}-\mathrm{u}) < (\mathrm{b}-\mathrm{a})$. (The g^{+} and g^{-} terms cancel.) This gives the same gross determination of dynamic controllability as the previous (Bhargava, Muise, and Williams 2018) procedure but differs in terms of the specific dynamic strategy. For the example, we get

$$\begin{tikzcd}
& \mathrm{X} \arrow[ld, "\mathrm{y}:1000"'] & \\
\mathrm{Y'} \arrow[rd, "-990"'] & & \mathrm{Y''} \arrow[lu, "Y:-5"'] \\
& \mathrm{Z} \arrow[ru, "0"'] &
\end{tikzcd}$$

and then, applying the usual STNU reductions, we end up with $\mathrm{X} \xrightarrow{10} \mathrm{Z}$ and $\mathrm{X} \xleftarrow{\mathrm{Y}:-5} \mathrm{Z}$ edges, which corresponds to a dynamic strategy of "Wait for Y until time 5 after X, and then execute Z before time 10 after X," which is the more flexible strategy we discussed earlier.

For the example where EZ has bounds [990,1000], the Y'Z and ZY" bounds are the only ones affected, and we get the situation depicted in the following figure.

$$\begin{tikzcd}
& \mathrm{X} \arrow[ld, "\mathrm{y}:1000"'] & \\
\mathrm{Y'} \arrow[rd, "0"'] & & \mathrm{Y''} \arrow[lu, "Y:-5"'] \\
& \mathrm{Z} \arrow[ru, "-990"'] &
\end{tikzcd}$$

Here, Y" observes the "Wait for Y until time 5 after X" condition, and then Z is released 990 units later. We will see later that the y:1000 bound on XY' can be interpreted as an upper bound of "Y or 1000 after X, whichever comes later," and then the same upper bound applies to Z. This strategy also matches our intuition.

These examples underscore our understanding that Y' and Y" are NOT the same timepoint as Y, although they are derived from it. The original Y timepoint in figure 2 has bounds of $[\mathrm{a}+\mathrm{g}^{-}, \mathrm{b}+\mathrm{g}^{+}]$ and these would need to be used, for example, if we were to consider placing requirements on Y itself. (This is apparently not within the scope of the Variable Delay formalism.)

In this section, to facilitate comparisons, we have used variable names that approximate those used in the Variable Delay paper. However, from now on we will adopt the convention, in most cases, of using bounds $[q^{-}, q^{+}]$ for any link whose endpoint is Q. We hope this will be useful as a mnemonic aid.

Single-Headed POSTNUs

The hidden timepoints in a POSTNU may be partitioned into separate groups whose elements are connected to each other by contingent links. A group is thus a connected component of the undirected graph obtained by *(i)* removing all requirement links from the POSTNU and *(ii)* replacing contingent links by their undirected variant. Since the STNU definition does not permit two contingent links to have the same endpoint, each group, together with an activation timepoint, will form a tree-like structure.

We now turn our attention to the special case where the hidden timepoints occur in groups consisting of linear chains of contingent links with a single non-hidden entrance and single non-hidden exit. We will call these *Single-Headed* POSTNUs.

For instance in a network $A \Rightarrow E_1 \Rightarrow E_2 \Rightarrow B$, E_1 and E_2 are hidden timepoints that belong to the same hidden group. (Notation convention: any E_i timepoint is hidden.)

Without loss of generality we may assume the entrance timepoint is controllable since otherwise it could be replaced by a controllable with a $[0,0]$ link to the original entrance. The exit timepoint is necessarily observable.

In this paper, **we exclude direct requirement links between two hidden timepoints**,[1] but otherwise the hidden timepoints (and entrance and exit) may participate in requirement links to other timepoints in the network. We then assume without loss of generality that timepoints directly linked to hidden timepoints are controllable, using $[0,0]$ link replacement if necessary.

As shown in figure 2, the Variable Delay problems may be regarded as a special case of Single-Headed POSTNUs, with limitations on the hidden groups and requirement links.

Analysis From First Principles

In our analysis we will first restrict our attention to simple Single-Headed POSTNUs, where the hidden groups each contain only one hidden timepoint, and later relax that restriction.

In an earlier section, we described a "doubling" strategy that enhanced the flexibility of execution. We now present

[1]For simplicity—the consequence of allowing them is unclear.

$$\begin{tikzcd}
\mathrm{X} \arrow[r, Rightarrow, "{[e^-, e^+]}"] & \mathrm{E} \arrow[r, Rightarrow, "{[y^-, y^+]}"] \arrow[d, "{[z^-, z^+]}"] & \mathrm{Y} \\
& \mathrm{Z} &
\end{tikzcd}$$

Figure 6: Generic Simple Problem

a first principles analysis in which the doubling arises naturally. The analysis focuses on mathematical equivalences that are independent of context. This eliminates some of the contextual restrictions that applied in the Variable Delay setting. As a side-benefit, the analysis sheds some additional light on the semantics of the upper-case and lower-case labeled edges used in the STNU work (Morris 2014).

For a POSTNU, we may divide the projections into groups that have the same values for their observable timepoints. We will call these groups *macro-projections*. The full projections that also specify the hidden timepoint values will be called *micro-projections*. Thus, each macro-projection consists of a set of micro-projections. In effect, each macro-projection, considered in isolation, may be regarded as a separate Strong Controllability problem whose projections are its micro-projections. Then each hidden timepoint will have a range of values within a particular macro-projection, and this range will depend on the values of the observables in the macro-projection.

For example, with the POSTNU (where E is hidden)

$$\mathrm{X} \xRightarrow{[0,10]} \mathrm{E} \xRightarrow{[0,10]} \mathrm{Y}$$

the macro-projection where XY = 15 consists of all the micro-projections where XE and EY sum to 15, such as $6 + 9$, $10 + 5$, etc. Within this set of micro-projections, E can range from 5 to 10 (after X). Notice while E can vary, the lower and upper bounds of the range, E_{lo} and E_{hi}, are fixed within the macro-projection. As we will see, their values can be expressed in terms of formulas involving the observables. Thus, we may regard them as virtual timepoints that live within the macro-projection, or *virtual observables* (although we only know their values *after* the relevant real observables have been observed). If we now impose a $[z^-, z^+]$ requirement on EZ, where Z is an executable timepoint, a worst-case analysis suggests we should enforce that by adding constraints $\mathrm{Z} \geq \mathrm{E}_{\mathrm{hi}} + z^-$ and $\mathrm{Z} \leq \mathrm{E}_{\mathrm{lo}} + z^+$. In contrast to the case for virtual observables, for the *executable* timepoint Z, we *do* need to know these constraints are satisfied by the time Z is scheduled.

Redirected Requirements We now consider this analysis in more detail for the simple generic problem shown in figure 6 (similar to Variable Delay). Here, X and Y are non-hidden timepoints. Both of these give us information bounding the occurrence of E as follows:

$$\begin{aligned}
\text{E-X} &\geq e^- \\
\text{E-X} &= \text{(Y-X)-(Y-E)} \\
&\geq \text{(Y-X)} - y^+
\end{aligned}$$

Thus,

$$\text{E-X} \geq \max(e^-, \text{(Y-X)} - y^+)$$

Similarly,

$$\begin{aligned}
\text{E-X} &\leq e^+ \\
\text{E-X} &= \text{(Y-X)} - \text{(Y-E)} \\
&\leq \text{(Y-X)} - y^-
\end{aligned}$$

so

$$\text{E-X} \leq \min(e^+, \text{(Y-X)} - y^-)$$

It is not hard to see that these are *tight* bounds; they represent the minimum and maximum extent of E-X within the macro-projection determined by X and Y. As discussed in the example, we will designate the lower and upper virtual observables by E_{lo} and E_{hi} respectively.

It is convenient to simplify the formulas by writing $\dot{Y}$ for Y-X. Thus, the X to E link has inferred bounds of

$$[\max(e^-, \dot{Y} - y^+), \min(e^+, \dot{Y} - y^-)]$$

Given a particular macro-projection, a dynamic strategy will need to specify a value for Z that works for *all* the associated micro-projections, i.e., for all the values of E within this range. Thus, we require $\mathrm{Z} - \mathrm{E} \geq z^-$ for each such E. It is not hard to see [2] that this is true **if and only if** it is true for the upper bound of the range, i.e., $\mathrm{Z} - \mathrm{E}_{\mathrm{hi}} \geq z^-$. Similarly, the lower-bound requirement is **equivalent** to $\mathrm{Z} - \mathrm{E}_{\mathrm{lo}} \leq z^+$.

We can rewrite these requirements as supplying a lower bound for XZ of $\mathrm{E}_{\mathrm{hi}} + z^-$ or

$$\min(e^+ + z^-, \dot{Y} + z^- - y^-)$$

and an upper bound of $\mathrm{E}_{\mathrm{lo}} + z^+$ or

$$\max(e^- + z^+, \dot{Y} + z^+ - y^+)$$

Notice the min/max modifiers have become reversed with respect to the lower and upper bounds. One consequence is that the bounds now represent implicit disjunctions rather than implicit conjunctions. However, we will see that the two alternatives can be processed together in a way that avoids an exponential blowup.

Observability Tightening These derived bounds may not be directly observable. For example, $(z^- - y^-)$ may be negative in which case the value of $\dot{Y} + (z^- - y^-)$ is unknown until the later time when (Y-X) is actually observed. If $(z^- - y^-)$ is non-negative, then $\dot{Y} + (z^- - y^-)$ is observable and can be left unchanged. Otherwise, when executing Z we must replace $\dot{Y} + (z^- - y^-)$ by the observable $\dot{Y}$, which gives a strictly tighter lower bound that guarantees the actual bound will be satisfied. We call this process *observability tightening*. It is important to note that we only apply it to *executable* timepoints, which is where the dynamic strategy applies.

When executing Z, the upper-bounds also need "observability tightening" but the process is different because of the asymetry of observation with respect to time. For example, if $(z^+ - y^+)$ is negative, then the strictly tighter bound derived

[2] $\mathrm{Z} - \mathrm{E} \geq z^-$ for each E, implies $\mathrm{Z} - \mathrm{E}_{\mathrm{hi}} \geq z^-$. Conversely, if $\mathrm{Z} - \mathrm{E}_{\mathrm{hi}} \geq z^-$ then $\mathrm{Z} - \mathrm{E} \geq \mathrm{Z} - \mathrm{E}_{\mathrm{hi}} \geq z^-$ for each E.

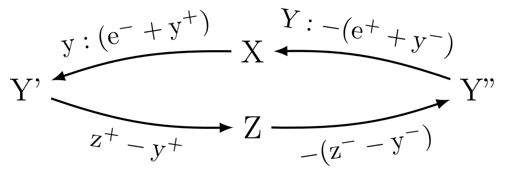

Figure 7: Distance Graph

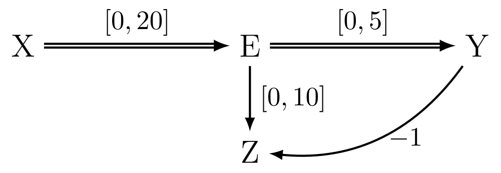

Figure 8: Cross Requirement Example

from $\dot{Y} + (z^+ - y^+)$ is "minus infinity", which is equivalent to dropping the $\dot{Y} + (z^+ - y^+)$ term from the max expression. If $(z^+ - y^+)$ is non-negative, then the term can be left unchanged.

Derived Observables Rather than interpreting the bounds on Z directly, we will pursue an alternative approach here, and decompose them by introducing intermediates with respect to the E_{lo} and E_{hi} values. Although E_{lo} and E_{hi} are only *virtual* observables whose values may not be known until later, we can form real observables from them by adding approriate delay terms. For example, $E_{hi} + y^- = \min(e^+ + y^-, \dot{Y})$ corresponds to an observation of "Y or $e^+ + y^-$ after X, whichever is earlier," and $E_{lo} + y^+ = \max(e^- + y^+, \dot{Y})$ may be paraphrased as "Y or $e^- + y^+$ after X, whichever is later." We will designate these derived observables as Y" and Y', respectively.

This motivates us to expand the lower bound for Z as

$$\min(e^+ + y^-, \dot{Y}) + (z^- - y^-)$$

and the upper bound as

$$\max(e^- + y^+, \dot{Y}) + (z^+ - y^+).$$

We will identify $-\min(u, \dot{Y})$ with the upper-case labeled weight Y:$-u$ and $\max(v, \dot{Y})$ with the lower-case labeled weight y:v, as used in an STNU labeled distance graph. (Morris 2014). (This will be justified later, but note that semantically, $\min(u, \dot{Y})$ is the same as the *Wait for Y until u after X* condition in an STNU.)

Introduction of the intermediate Y' and Y" thus produces the labeled distance graph shown in figure 7. This may be compared with figure 5.

Example We reiterate that the Y timepoint is distinct from the added Y' and Y" timepoints. The correlation between them is captured by the labeled weights in the distance graph. Consider, for example, the network shown in figure 8. If the YZ edge was not there, the network would be Dynamically Controllable, since Z could be executed between 0 and 5 after Y is observed. However, the YZ edge prevents that strategy by requiring Z to come before Y, so the full network is not Dynamically Controllable. The distance graph after the transformations is shown in figure 9. Notice the Lower-Case reduction applied to XYZ produces an XZ edge of weight -1, which then forms a semi-reduced negative cycle with the ZY"X path.

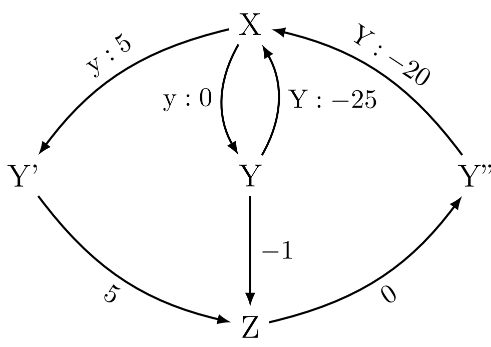

Figure 9: Cross Distance Graph

Hidden Timepoint Elimination After the requirement edges between Z and E are replaced by the corresponding edges between Z and X, E will be free of "side" links. At that point, the XE link can be composed with the EY link, giving a combined contingent link of XY with bounds $[e^- + y^-, e^+ + y^+]$, and E can be eliminated.

Now we return to the general case where there is a chain of hidden timepoints

$$X \Rightarrow E_1 \Rightarrow \ldots \Rightarrow E_n \Rightarrow Y$$

between X and Y. Consider the first timepoint E_1. The analysis that produced E_{lo} and E_{hi} depended only on knowing the contingent bounds for XE and EY. Viewing E_1 as if it were E, we know the bounds for XE directly, and we can compute bounds for EY by composing the contingent links in the $E_1 \Rightarrow \ldots \Rightarrow$ Y path. [3] We can then proceed as in the single E case to eliminate E_1. This process can be repeated with the other hidden timepoints in the chain until they are all eliminated.

At this point, what remains is a labeled distance graph with no hidden timepoints, which is a form suitable for input to a standard cubic Dynamic Controllability checking algorithm for STNUs (Morris 2014). This leads us to the following theorem.

Theorem 1 *For the given class of Single-Head POSTNUs, the transformation process followed by the standard cubic Dynamic Controllability checking algorithm provides a complete decision procedure.*

Proof: We have seen that the first transformation step replaces the original requirement constraints with equivalent ones. Because of the equivalence, this necessarily leaves the set of valid dynamic strategies unchanged. The observability tightening step does restrict the network, but any strategies eliminated by the step would be non-dynamic since they

[3] Requirements do not affect the domains of contingent links.

would depend on unobserved values. Thus, the set of dynamic strategies before and after the transformation process is the same. (This may be empty if the network is not dynamically controllable.)

Next we justify the identification of the max/min expressions with the labeled weights by showing they behave the same with respect to the key reductions used by the Dynamic Controllability checking algorithm. In the following, we assume $u > 0$ and $v \geq 0$ and $W \neq Y$.

Upper-Case Reduction
$-\min(u, \dot{Y}) + v \quad = -\min(u - v, \dot{Y} - v)$
$\quad = -\min(u - v, \dot{Y})$

Lower-Case Reduction
$-u + \max(v, \dot{Y}) \quad = \max(v - u, \dot{Y} - u)$
$\quad = v - u$

Cross-Case Reduction
$-\min(u, \dot{W}) + \max(v, \dot{Y}) \quad = v - \min(u, \dot{W})$
$\quad = -\min(u - v, \dot{W})$

Label Removal
$-\min(-v, \dot{Y}) \quad = v$

Note the use of the applicable **observability tightening** in the Upper and Lower cases. The Cross-Case reduction applies the Lower and Upper derivations in succession. [4] Label Removal follows from min simplification since $\dot{Y} \geq 0$.

The theorem then follows from the completeness of the (Morris 2014) algorithm. □

Note that this result extends and unifies the previous classes of POSTNU for which complete and tractable decision procedures are known (Bit-Monnot, Ghallab, and Ingrand 2016; Bhargava, Muise, and Williams 2018).

Multi-Headed Observations

In the previous sections, we discussed problems where bounds on the occurrence of a hidden timepoint could be inferred from a single observation. In this section, we consider the combined effect of multiple relevant observations, the so-called "Multi-Headed Problem," where there can be multiple determinations that a hidden event has occurred, each of which has its own bounds. This is illustrated in figure 10 for a two-headed problem.

In the figure, X and Z are executable timepoints, while E is hidden, and Y and W are observables. The past occurrence of E can be inferred from an observation of either Y or W, which also provide (different) bounds on when E occurred.

In this section, we present some partial results concerning these kinds of problems, and some possible approaches. A complete solution remains a challenge for future work.

Local Dynamic Controllability

Although the ultimate goal is to develop transformation methods that apply to POSTNU fragments independent of context, a useful first step is to characterize Dynamic Controllability for certain fragments in isolation. These can be used as a check on the validity of more general approaches,

[4]Hint: first apply the LC derivation using $u' = \min(u, \dot{W})$.

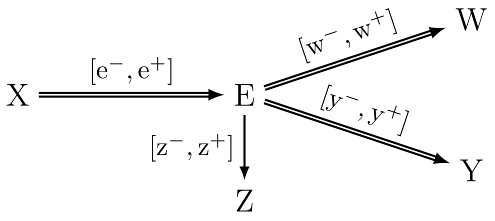

Figure 10: Two-Headed Problem

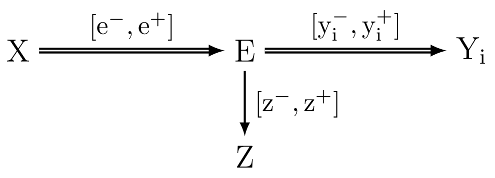

Figure 11: Multi-Headed Problem

and thus help to guide further research. Here we prove a result of this kind. It also illustrates some distinctions between the problem of checking Dynamic Controllability, and aspects regarding flexibility of execution.

The theorem applies to a problem with any number of "heads." Consider the generic example in figure 11 where i is repeated from 1 to n. Here X and Z are executable timepoints, while E is hidden, and the Y_i are separate observables.

Theorem 2 *The network in figure 11 is Dynamically Controllable if and only if* **either** *slack(EZ) $\geq$ slack(XE)* **or** *slack(EZ) $\geq$ slack(EY$_i$) for some i such that $z^+ \geq y_i^+$.*

Proof: If slack(EZ) $\geq$ slack(XE), then $z^+ + e^- \geq z^- + e^+$ and then executing Z in a way that satisfies XZ = $[z^- + e^+, z^+ + e^-]$ constitutes a dynamic strategy since the EZ requirement will be satisfied no matter the outcome of the XE contingent link. Note that XZ can be placed as a requirement even if $z^+ + e^-$ and $z^- + e^+$ are negative since X is controllable.

Also, if slack(EZ) $\geq$ slack(EY$_i$) and $z^+ \geq y_i^+$ for some i, then $z^+ - y_i^+ \geq z^- - y_i^-$, and executing Z in a way that satisfies Y$_i$ Z = $[z^- - y_i^-, z^+ - y_i^+]$ will constitute a dynamic strategy since then the EZ requirement will be satisfied no matter the outcome of the EY$_i$ contingent link. Note Z can be scheduled to satisfy this Y$_i$Z since $z^+ \geq y_i^+$ and Y$_i$ is observable.

Thus, the "if" direction is satisfied.

Conversely, suppose slack(EZ) < slack(XE) and, for all i, either slack(EZ) < slack(E Y$_i$) or $z^+ < y_i^+$.

Let Q be the set of i such that slack(EZ) < slack(E Y$_i$). Define q = min {slack(E Y$_i$) : i in Q}. Then slack(EZ) < q.

We define two projections P1 and P2 as follows.

- In P1, XE has its maximum extent e^+
- In P2, XE has extent $e^+ - q$
- For $i \in$ Q:
  - In P1, EY_i has its minimum extent y_i^-

  - In P2, EY_i has extent $y_i^- + q$
- For $i \notin$ Q:
  - For both P1 and P2, Y_i has its maximum extent y_i^+

Note that for $i \in$ in Q, the observed XY$_i$ = XE + EY$_i$ has the same extent $e^+ + y_i^-$ in both P1 and P2.

Also note that for $i \notin$ Q, $z^+ < y_i^+$, so in both P1 and P2, none of the Y$_i$ will have been observed by the time Z reaches its upper bound, and must have been scheduled.

From the above we see that P1 and P2 cannot be distinguished by any dynamic strategy, so Z must be scheduled at the same time in both projections.

Finally, we note that the value of E in P1 and P2 differs by an amount q. But slack(EZ) $<$ q and Z is fixed. It follows that the EZ constraint must be violated in either P1 or P2, which contradicts the assumption of a dynamic strategy. Thus the network is not Dynamically Controllable, proving the "only if" direction. $\square$

We remark that Theorem 2 is consistent with the Variable Delay transformations, as well as the doubling strategy, with respect to the determination of Dynamic Controllability. A point of interest is that for determining Dynamic Controllability, the property of importance is the existence of at least one "head" (or activation "tail") with less slack (i.e., uncertainty) than the requirement. However, we have seen that flexibility of execution can be enhanced by opportunistic use of observations whose slack may exceed that of the requirement.

Global Dynamic Strategy

In this section, we explore a first principles approach similar to that used in the single-headed case, and see what additional issues arise. In particular, we consider the two-headed problem in figure 10.

In this two-headed example, the observables are X, Y, and W. Each of the observations gives us information bounding the occurrence of E. We can then derive overall bounds in a manner similar to that used in the single-head case. This results in the following inferred bounds for the X to E link. (Recall that $\dot{Y}$ abbreviates Y-X, and $\dot{W}$ abbreviates W-X.)

$$[\max(e^-, \dot{Y} - y^+, \dot{W} - w^+), \min(e^+, \dot{Y} - y^-, \dot{W} - w^-)]$$

As for the single-head case, we define virtual observables E$_{\text{lo}}$ and E$_{\text{hi}}$ in terms of these bounds, and add constraints Z $\geq$ E$_{\text{hi}} + z^-$ and Z $\leq$ E$_{\text{lo}} + z^+$ to give an X to Z link with lower bound

$$\min\{e^+ + z^-, \dot{Y} + (z^- - y^-), \dot{W} + (z^- - w^-)\}$$

and upper bound

$$\max\{e^- + z^+, \dot{Y} + (z^+ - y^+), \dot{W} + (z^+ - w^+)\}.$$

Observability tightening must again be applied since Z is an executable timepoint. One difference from the single-head case is that there are two observables in each bound, and the tightening needed may be different in each case. In particular, the upper-bound tightening (which potentially drops terms from the max expression) may eliminate one or both of the observable terms. If it eliminates both, this leaves a single value, which is analogous to an application of the Lower Case Reduction. If it eliminates only one, this leaves what is effectively a single-head expression. It may also drop no terms, leaving an expression with multiple observable terms.

At this point it is unclear how far the analogy to the single-head case can be carried further. Considering just the lower-bound expression, the values $(z^- - y^-)$ and $(z^- - w^-)$ added to the Y and W observables may be different, so there is no one term that we can "take outside," leaving a "bare" observable, as we did for the single-headed case. This makes the approach of introducing intermediate observables unclear, and even if we did, the double-observable expressions cannot be identified with conventional STNU labels. [5]

However, if we could sidestep the problem of checking Dynamic Controllability, the multi-observation bounds on executable timepoints could in fact be interpreted in accordance with a dynamic strategy. For example, consider a lower bound of $\min(10, \dot{Y} + 5, \dot{W})$. (Note that after observability tightening, any quantities added to an observable will be non-negative.) This can be interpreted as an observation of "Y+5 or W or 5 after X, whichever is earlier." Similarly, an upper bound of $\max(20, \dot{Y}, \dot{W} + 10)$ corresponds to "Y or W+10 or 20 after X, whichever is later."

Closing Remarks

We have built on previous work in the area of STNUs, especially Variable Delay, and extended it to a POSTNU setting. By means of a detailed First Principles analysis, we have shown how to achieve a more flexible dynamic strategy for execution. The results provide for additional context in terms of network configuration. For the "single-headed" class of problems considered, the determination of Dynamic Controllability is complete and correct, and the dynamic strategy preserves the full flexibility. We have also explored multi-headed problems and presented partial results in this area.

Since the Dynamic Controllability and Strong Controllability problems for STNUs are tractable, and since POSTNUs are essentially a combination of the two, it is plausible to think that the general POSTNU problem might be tractable, although a general solution to this problem is unknown at the present time. It seems to be a difficult problem to analyze, but a very interesting one, in view of the "arrow of time" with respect to the observables, but not the unobservables. The Variable Delay paper may be regarded as establishing a beachhead in terms of new approaches to this problem, and the current paper makes further forays in this area. We are hopeful that future advances may lead to the sought-after general solution.

References

Bhargava, N.; Muise, C.; and Williams, B. 2018. Variable-delay controllability. In *International Joint Conference on Artificial Intelligence (IJCAI'18).*

[5] Of course, one could introduce an explicit disjunction and handle the observables separately, but that seems to abandon the search for a tractable algorithm.

Bit-Monnot; Ghallab, M.; and Ingrand, F. 2016. Which contingent events to observe for the dynamic controllability of a plan. In *International Joint Conference on Artificial Intelligence (IJCAI'16)*.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Hunsberger, L. 2009. Fixing the semantics for dynamic controllability and providing a more practical characterization of dynamic execution strategies. In *International Symposium on Temporal Representation and Reasoning (TIME'09)*.

Moffitt, M. D. 2007. On the partial observability of temporal uncertainty. In *AAAI Conference on Artificial Intelligence (AAAI'07)*.

Morris, P., and Muscettola, N. 2005. Dynamic controllability revisited. In *AAAI Conference on Artificial Intelligence (AAAI'05)*.

Morris, P.; Muscettola, N.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *International Joint Conference on Artificial Intelligence (IJCAI'01)*.

Morris, P. 2014. Dynamic controllability and dispatchability relationships. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR'14)*.

Vidal, T., and Fargier, H. 1999. Handling contingency in temporal constraint networks: from consistency to controllabilities. *Journal of Experimental & Theoretical Artificial Intelligence (JETAI)* 11:23–45.